

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Serial No.....09/675,688
Filing Date09/29/2000
Inventorship..... Ashraf A. Michail
Applicant..... Microsoft Corporation
Group Art Unit2178
Examiner..... Kyle R. Stork
Attorney's Docket No.MS1-675US
Title: Selection Services Component For An Extensible Editor

SUPPLEMENTAL APPEAL BRIEF

To: Commissioner for Patents
PO Box 1450
Alexandria, Virginia 22313-1450

From: Lance Sadler (Tel. 509-324-9256x226; Fax 509-323-8979)
Customer No. 22801

Pursuant to 37 C.F.R. §41.37, Applicant hereby submits this supplemental appeal brief for application 09/675,688, filed September 29, 2000, within the requisite time from the date of filing the Notice of Appeal. This supplemental appeal brief is filed responsive to a communication from the Office dated June 15, 2006 entitled "Notification of Non-Compliant Appeal Brief". The supplemental appeal brief corrects the formatting issues and should replace the previously-filed brief. Accordingly, Applicant appeals to the Board of Patent Appeals and Interferences seeking review of the Examiner's rejections.

<u>Appeal Brief Items</u>	<u>Page</u>
(1) Real Party in Interest	3
(2) Related Appeals and Interferences	3
(3) Status of Claims	3
(4) Status of Amendments	3
(5) Summary of Claimed Subject Matter	4
(6) Grounds of Rejection to be Reviewed on Appeal	5
(7) Argument	5
(8) Appendix of Appealed Claims	20
(9) Evidence appendix	21
(10) Related Proceedings appendix	22

(1) Real Party in Interest

The real party in interest is Microsoft Corporation, the assignee of all right, title and interest in and to the subject invention.

(2) Related Appeals and Interferences

Appellant is not aware of any other appeals, interferences, or judicial proceedings which will directly affect, be directly affected by, or otherwise have a bearing on the Board's decision to this pending appeal.

(3) Status of Claims

Claims 1-6, 23-28, and 36-41 stand rejected and are pending in the Application. Claims 1-6, 23-28, and 36-41 are set forth in the Appendix of Appealed Claims on page 16.

(4) Status of Amendments

No claims were amended subsequent to final rejection.

(5) Summary of Claimed Subject Matter

A concise explanation of each of the independent claims is included in this Summary section, including specific reference characters. These specific reference characters are examples of particular elements of the drawings for certain embodiments of the claimed subject matter and the claims are not limited to solely the elements corresponding to these reference characters.

With regard to claim 1, in an extensible electronic document editor, a selection services component comprises a selection services interface that provides one or more methods to enable an editor extension to override a selection function provided by the electronic document editor and provide a customized model for the selection function, (Page 13, Lines 9-20; Fig. 2) and to clear, add or remove a segment from a selection object using the customized model for the selection function (Page 28, lines 3-11).

With regard to claim 23, a computer-readable medium has computer-executable instructions for one or more interfaces that, when executed by an extensible electronic document editor on a computer, perform the following steps: override a selection function provided by the electronic document editor to provide a customized model for the selection function (Page 13, Lines 9-20; Fig. 2); receive parameters from a designer through a selection services interface; utilize the parameters to clear, add or remove a selected segment from a selection object of an electronic document using the customized model for the selection function (Page 28, lines 3-11); return values to the designer regarding the state of the selected segment (Page 21, lines 18-23).

With regard to claim 36, a method for providing selection services to one or more extensions in an extensible editor comprises receiving a request from an extension to utilize a selection services component; facilitating the request by presenting a selection services interface that is accessible by the extension and that overrides a selection function provided by the extensible editor to provide a customized model for the selection function (Page 13, Lines 9-20; Fig. 2); and communicating with the extension through the selection services interface to

enable the extension to clear, add, or remove a selected segment from a selection object of an electronic document using the customized model for the selection function (Page 28, lines 3-11).

(6) Grounds of Rejection to be Reviewed on Appeal

Claims 1-6, 23-28, and 36-41 stand rejected under 35 U.S.C. § 103(a) as being obvious over Walkowski (10 Minute Guide to WordPerfect 5.1 for Windows, 1992) (hereinafter “Walkowski”), in view of U.S. Patent No. 6,289,450 to Pensak et al. (hereinafter “Pensak”).

(7) Argument

A. The rejections under 35 U.S.C. §103(a) over Walkowski and Pensak do not establish a *prima facie* case of obviousness because the combination does not teach or suggest all the claim features.

In the discussion in this section that follows, a section entitled “Applicant’s Disclosure” is provided and describes aspects of Applicant’s disclosure that are germane to the claimed embodiments. Following this, sections entitled “The Walkowski Reference” and “The Pensak Reference” are provided and respectively describe these references. Following these sections, the individual claims are discussed under their own respective headings.

Applicant’s Disclosure

Applicant’s disclosure describes an extensible editor for editing electronic documents and/or content. The extensible editor provides a selection services component that allows extensions to interface with a selection model of the editor

and provide a custom selection process to the editor. The editor can thereby implement the custom selection process without knowing details of how the selection process is implemented in the extension.

Perhaps a good place to start a discussion of Applicant's disclosure is with the overview provided by Fig. 2.

Fig. 2 illustrates an extensible editor 206 that includes an event routing controller 208, a designer extensibility mechanism 210, a selection services component 212, and a highlight rendering services component 214. Three designers 216, 218, 220 are also stored in the memory 204. Each of the designers 216 - 220 communicates with the editor 206 via the designer extensibility mechanism 210. Each designer 216 – 220, as shown, also communicates with the selection services component 212 and the highlight rendering component 214. Each designer 216 – 220 attaches to the editor 206 through the designer extensibility mechanism 210.

A designer is an editor extension that is used to extend the functionality of the editor 206 and to customize the behavior of the editor 206.

The designers 216 - 220 work by intercepting events and commands occurring in, or received by, the editor 206. When one or more of the designers 216 - 220 intercepts an event (or command), the designer can change how the editor 206 handles the event. Generally, a designer is written to either supplement or override the editor's behavior. Several designers may be attached to the editor 206 at once, thereby dynamically enabling multiple levels of custom functionality.

Designers offer a very powerful tool for customizing the editor 206. Virtually any part of the editor's behavior can be changed. For example, designers

may be used to add spell checking capability to the editor 206, to add table editing functionality, to add annotation or revision-tracking capability, and so on.

The designer extensibility mechanism 210, the selection services component 212, and the highlight rendering services 214 of the editor 206 shown in Fig. 2 provide specific functionality to the editor 206.

With regard to the selection services component 212, perhaps a good starting place is on page 27 of Applicant's disclosure and its corresponding discussion of Fig. 6.

Selection services provide extensions a way to modify a selection process of an extensible editor to which the designers are coupled. Fig. 6 is a block diagram of an extensible editor 600 that includes a designer interface 602, an event routing mechanism 604, and a selection services component 606. The selection services component 606 includes several interfaces: a selection services interface 608 (ISelectionServices), a selection services listener interface 610 (ISelectionServiceListener), an element segment interface 612 (IElementSegment), a segment list interface 614 (ISegmentList), and a segment interface 616 (ISegment).

The role of the selection services interfaces 608 – 616 is to provide designers or other editing extensions with the ability to modify the logical selection state. Consequently, all editing commands and services can interact with a custom selection model without having detailed knowledge of the designer that is implementing the selection.

For example, the “bold” command is able to implement the operation of making something bold without having any knowledge of the specifics of a given

designer. The command is only aware of what part of the document is selected, and it is configured to make the selected region of the document bold.

Selection Service Interface (ISelectionServices)

The selection services interface 608 provides methods to programmatically clear, add and remove segments from a selection object. The methods include an add element segment method 618 (AddElementSegment), a get markup container method 620 (GetMarkupContainer), a get selection services listener method 622 (GetSelectionServicesListener), an add segment method 624 (AddSegment), a remove segment method 626 (RemoveSegment), and a set selection type method 628 (SetSelectionType). These methods are described in detail starting on page 28.

Accordingly, the extensible editor provides a selection services component that allows extensions to interface with a selection model of the editor and provide a custom selection process to the editor. The editor can thereby implement the custom selection process without knowing details of how the selection process is implemented in the extension.

The Walkowski Reference

Walkowski is entitled "Guide to WordPerfect 5.1 For Windows: Quick Lessons for WordPerfect for Windows Success."

Walkowski teaches exactly what its title suggests: how to use a word processor. This reference is really only a general treatment of how one uses WordPerfect 5.1. For example, this reference includes sections entitled "Using the

Button Bar” which steps the reader through how one uses a button bar; “Exiting WordPerfect” which steps the reader through how one exits WordPerfect; “Editing your Text” which steps a reader through how one can edit text in WordPerfect; and “Deleting, Moving, and Copying Blocks of Text” which steps a reader through how one can operate on text using WordPerfect.

The Pensak Reference

Pensak pertains generally to information security architecture for encrypting documents for remote access while maintaining access control.

Pensak provides for encrypting electronic information, such as a document, so that only users with permission may access the document in decrypted form. The process of encrypting the information includes selecting a set of policies as to who may access the information and under what conditions. A remote server stores a unique identifier for the information and associates an encryption/decryption key pair and access policies with the information. Software components residing on the author's computer retrieve the encryption key from the remote server, encrypt the information, and store the encrypted information at a location chosen by the author. A user wishing to access the information acquires the encrypted information electronically. Software components residing on the viewing user's computer retrieve the associated decryption key and policies, decrypt the information to the extent authorized by the policies, and immediately delete the decryption key from the viewing user's computer upon decrypting the information and rendering the clear text to the viewing user's computer screen.

The software components are also capable of prohibiting functional operations by the viewing user's computer while the clear text is being viewed.

The Claims

Claims 1-6

Claim 1 recites in an extensible electronic document editor, a selection services component comprising a selection services interface that provides one or more methods to enable an editor extension to override a selection function provided by the electronic document editor and provide a customized model for the selection function, and to clear, add or remove a segment from a selection object using the customized model for the selection function.

In order to make out a § 103(a) rejection, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974).

In making out the rejection of claim 1, the Office argues that the combination of Walkowski and Pensak renders this claim obvious. Specifically the Office claims that Walkowski teaches “an electronic document editor, a selection services component comprising a selection services interface that provides one or more methods to enable an editor extension to clear, add or remove a segment from a selection object.” The Applicant disagrees and respectfully traverses the Office’s rejection.

Walkowski does not in any way teach or suggest an “*editor extension*” to clear, add or remove a segment from a selection object.” First, there is no mention in Walkowski of an “editor extension.” As discussed in the Applicant’s specification, an editor extension, or designer, is an extension that is “used to

extend functionality of [an] editor and to customize the behavior of the editor” (Applicant’s Specification, page 12). Walkowski does not teach or in any way suggest the use of an editor extension. Instead, Walkowski is only concerned with the editor itself -- WordPerfect.

Second, Walkowski does not teach or in any way suggest the ability to “*clear, add or remove a segment from a selection object.*” The Office argues that pages 23-25 and 31-32 of Walkowski teach this subject matter. However, these pages of Walkowski teach editing text, inserting text, typing over existing text, deleting text, undeleting text, and moving text in a WordPerfect document. All of these teachings are basic word processing maneuvers. Walkowski does not in any way teach or suggest the ability to “*clear, add or remove a segment from a selection object.*” In fact, it appears as though the Office has simply conducted a key word search in order to reject this claim under Walkowski. For at least these reasons, the Office has failed to make out a *prima facie* case of obviousness.

Additionally, the Office admits that Walkowski fails to specifically disclose an editor extension to override a selection function provided by the electronic document editor and provide a customized model for the selection function. However, the Office then argues that Pensak teaches this subject matter and that it would have been obvious to combine Walkowski with Pensak. The Applicant disagrees and traverses the Office’s argument.

Pensak does not in any way teach or suggest the ability of an editor extension to override a selection function provided by the electronic document editor and provide a customized model for the selection function. In order to assist

the Office in understanding why Pensak does not teach this subject matter, the following excerpt from the Applicant's specification is reproduced below:

Page 13, Lines 9-20 of Applicant's Specification

The designers 216 - 220 work by intercepting events and commands occurring in, or received by, the editor 206. When one or more of the designers 216 - 220 intercepts an event (or command), the *designer can change how the editor 206 handles the event. Generally, a designer is written to either supplement or override the editor's behavior.* Several designers may be attached to the editor 206 at once, thereby dynamically enabling multiple levels of custom functionality.

Designers offer a very powerful tool for customizing the editor 206. Virtually any part of the editor's behavior can be changed. *For example, designers may be used to add spell checking capability to the editor 206, to add table editing functionality, to add annotation or revision-tracking capability, and so on.* It is noted that, although only three designers 216 - 220 are shown in conjunction with the editor 206, any number of designers may be connected to the editor 206.

Pensak teaches the ability of an administrator to "control the functions available to a particular authoring user, which might depend on the fees paid by the authoring user" (Pensak column 4, lines 19-21). In other words, Pensak teaches the ability to restrict certain users from using certain functions associated with an electronic document, such as the ability to copy or print text. There is no mention whatsoever in Pensak of the ability to *override* a selection function provided by the electronic document editor and *provide a customized model* for the selection function.

The Office, in its Advisory Action dated November 2, 2005 argues that "although the customized tool may restrict original functions of the authoring tool, such as not allowing unauthorized users to print, save, or view a complete

document, *this does not preclude the customized tool from providing a user customized function.*” The Applicant strongly disagrees with this assertion. If Pensak truly does provide a customized model for the selection function, then the Applicant urges the Office to refer the Applicant to the section of Pensak that discloses this subject matter. The Office has failed to do this, and the Applicant submits this is due to the fact that Pensak does not in any way teach or suggest the ability to provide a customized model for the selection function. As such, and for this additional reason, the Office has failed to make out a *prima facie* case of obviousness.

Accordingly, for all of the reasons discussed above, this claim is allowable.

Claims 2-6 depend from claim 1 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 1, are neither disclosed nor suggested in the references of record, either singly or in combination with one another. Moreover, the failure of the Office to make out a *prima facie* case of obviousness with regard to the rejection of claim 1 is also fatal with regard to the Office’s attempt to make out a *prima facie* case of obviousness of these claims.

Claims 23-28

Claim 23 recites a computer-readable medium having computer-executable instructions for one or more interfaces that, when executed by an extensible electronic document editor on a computer, perform the following steps:

- override a selection function provided by the electronic document editor to provide a customized model for the selection function;

- receive parameters from a designer through a selection services interface;
- utilize the parameters to clear, add or remove a selected segment from a selection object of an electronic document using the customized model for the selection function;
- return values to the designer regarding the state of the selected segment.

In making out the rejection of this claim, the Office argues that “claim 23 reflects similar subject matter claimed in claim 1 and is rejected along the same rationale” (August 12, 2005 Office Action). Accordingly, for the same reasons as discussed with regards to claim 1, (excluding, of course, the argument with regard to claim 1’s specifically recited subject matter), the Applicant submits that the Office has failed to make out a *prima facie* case of obviousness.

Additionally, claim 23 recites subject matter that is not found in claim 1 and which has not been examined by the Office. The Applicant submits that the combination of Walkowski and Pensak does not teach or in any way suggest a computer-readable medium having computer-executable instructions for one or more interfaces that, when executed by an extensible electronic document editor on a computer, perform the following steps: *receive parameters from a designer through a selection services interface; and return values to the designer regarding the state of the selected segment*. For this additional reason, the Office has failed to make out a *prima facie* case of obviousness.

Accordingly, for all of the reasons discussed above, this claim is allowable.

Claims 24-28 depend from claim 23 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 23, are neither disclosed

nor suggested in the references of record, either singly or in combination with one another.

Claims 36-41

Claim 36 recites a method for providing selection services to one or more extensions in an extensible editor, the method comprising:

- receiving a request from an extension to utilize a selection services component;
- facilitating the request by presenting a selection services interface that is accessible by the extension and that overrides a selection function provided by the extensible editor to provide a customized model for the selection function; and
- communicating with the extension through the selection services interface to enable the extension to clear, add, or remove a selected segment from a selection object of an electronic document using the customized model for the selection function.

In making out the rejection of this claim, the Office uses a similar argument as was used in making out a rejection of claim 1. Accordingly, for the same reasons as discussed with regards to claim 1 (excluding, of course, the argument with regard to claim 1's specifically recited subject matter), the Applicant submits that the Office has failed to make out a *prima facie* case of obviousness.

Accordingly, for all of the reasons discussed above, this claim is allowable.

Claims 37-41 depend from claim 36 and are allowable as depending from an allowable base claim. These claims are also allowable for their own recited features which, in combination with those recited in claim 36, are neither disclosed nor suggested in the references of record, either singly or in combination with one another.

B. The rejections under 35 U.S.C. §103(a) over Walkowski and Pensak do not establish a prima facie case of obviousness because the Office has used hindsight reconstruction to combine Walkowski with Pensak.

In making out a § 103(a) rejection, there is a requirement that there must be some reason, suggestion, or motivation *from the prior art*, as a whole, for the person of ordinary skill to have combined or modified the references. *See, In re Geiger*, 2 USPQ 2d 1276, 1278 (Fed. Cir. 1987). It is impermissible to use the claimed invention as an instruction manual or “template” to piece together the teachings of the prior art so that the claimed invention is rendered obvious. One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention. *In re Fritch*, 23 USPQ 2d 1780, 1784 (Fed. Cir. 1992). In this case, the Office has clearly used hindsight reconstruction to combine Walkowski with Pensak.

Walkowski is a quick guide for users who want to learn the basics of word processing on WordPerfect. Walkowski teaches simple word processing maneuvers such as editing text in a document using various word processing functions. Yet, the Office argues it would have been obvious to one of ordinary skill in the art at the time of the applicant’s invention to have combined Walkowski’s word processor with Pensak’s method to control access to a document, “since it would have allowed an administrator the capability to control access to documents and functions.”

This combination is misplaced at best because there is nothing in Walkowski that suggests the desirability of controlling access to documents and functions. As such, it is clear that the Office has used hindsight reconstruction in

piecing together Walkowski and Pensak. Accordingly, for at least this reason, the Office has failed to make out a *prima facie* case of obviousness.

C. The rejections under 35 U.S.C. §103(a) over Walkowski and Pensak do not establish a *prima facie* case of obviousness because the modification suggested by the Office would render Walkowski unsatisfactory for its intended purpose.

The Office, in its Office Action dated August 12, 2005, argues that the combination of Walkowski and Pensak renders all of the Applicant's claims obvious under 35 U.S.C. §103(a). The Office submits that "it would have been obvious to one of ordinary skill in the art at the time of the Applicant's invention to have combined Walkowski's editor with Pensak's editor, since it would have allowed an administrator the capability to control access to documents and functions."

The Applicant submits that the Office has not established a *prima facie* case of obviousness because the combination of Walkowski with Pensak would render Walkowski unsatisfactory for its intended purpose. A modification proposed by the Office cannot render the reference unsatisfactory for its intended purpose. Specifically, MPEP §2143.01 entitled "Suggestion or Motivation To Modify the References" instructs as follows:

THE PROPOSED MODIFICATION CANNOT RENDER THE PRIOR ART UNSATISFACTORY FOR ITS INTENDED PURPOSE

If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. In re Gordon, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984) (Claimed device was a blood filter assembly for use during medical procedures wherein both

the inlet and outlet for the blood were located at the bottom end of the filter assembly, and wherein a gas vent was present at the top of the filter assembly. The prior art reference taught a liquid strainer for removing dirt and water from gasoline and other light oils wherein the inlet and outlet were at the top of the device, and wherein a pet-cock (stopcock) was located at the bottom of the device for periodically removing the collected dirt and water. The reference further taught that the separation is assisted by gravity. The Board concluded the claims were *prima facie* obvious, reasoning that it would have been obvious to turn the reference device upside down. The court reversed, finding that if the prior art device was turned upside down it would be inoperable for its intended purpose because the gasoline to be filtered would be trapped at the top, the water and heavier oils sought to be separated would flow out of the outlet instead of the purified gasoline, and the screen would become clogged.).

MPEP § 2143.01

Walkowski is entitled “Guide to WordPerfect 5.1 For Windows: Quick Lessons for WordPerfect for Windows Success.” Walkowski teaches exactly what its title suggests: how to use a word processor. This includes such lessons as “Using the Button Bar”; “Exiting WordPerfect”; “Editing your Text”; and “Deleting, Moving, and Copying Blocks of Text.”

Pensak pertains generally to information security architecture for encrypting documents for remote access while maintaining access control. Pensak enables an “authoring user to block certain functions normally accessible by the viewing user. For example, the authoring user may deny a viewing user privileges such as printing and copying of the clear text” (Pensak, Column 2, Lines 24-28). Thus, if Pensak were combined with Walkowski, many of the functions described in Walkowski (such as editing and copying text) could be blocked. This would render Walkowski unsatisfactory for its intended purpose which is to enable users to use simple word processing software where the simple functionality of the word processor is enabled. Thus, the combination suggested by the Office would render

Walkowski unsatisfactory for its intended purpose. Accordingly, for at least this reason, the Office has failed to make out a *prima facie* case of obviousness.

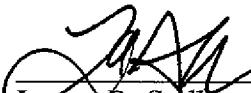
Conclusion

The Office has not established a *prima facie* case of obviousness for the reasons noted above. Accordingly, Applicant respectfully requests that the rejections be overturned and that the pending claims be allowed to issue.

Respectfully Submitted,

Dated: 6/26/06

By:



Lance R. Sadler
Lee & Hayes, PLLC
Reg. No. 38,605
(509) 324-9256 ext. 226

(8) Appendix ofAppealed Claims

1. (Previously Presented) In an extensible electronic document editor, a selection services component comprising a selection services interface that provides one or more methods to enable an editor extension to override a selection function provided by the electronic document editor and provide a customized model for the selection function, and to clear, add or remove a segment from a selection object using the customized model for the selection function.

2. (Original) The selection services component as recited in claim 1, wherein one of the methods provided further comprises a method that adds an element segment to an editable selection.

3. (Previously Presented) The selection services component as recited in claim 1, wherein the selection services interface is adapted to enable the extensible electronic document editor to interact with the customized selection model without exposing details of the editor extension.

4. (Original) The selection services component as recited in claim 1, wherein one of the methods provided further comprises a method that adds a segment to an editable selection.

5. (Original) The selection services component as recited in claim 1, wherein one of the methods provided further comprises a method that removes a segment from an editable selection.

6. (Original) The selection services component as recited in claim 1, wherein one of the methods provided further comprises a method that sets a selection type.

7.-22. (Canceled)

23. (Previously Presented) A computer-readable medium having computer-executable instructions for one or more interfaces that, when executed by an extensible electronic document editor on a computer, perform the following steps:

override a selection function provided by the electronic document editor to provide a customized model for the selection function;

receive parameters from a designer through a selection services interface;

utilize the parameters to clear, add or remove a selected segment from a selection object of an electronic document using the customized model for the selection function;

return values to the designer regarding the state of the selected segment.

24. (Original) The computer-readable medium as recited in claim 23, wherein the utilizing the parameters further comprises utilizing the parameters to add an element segment to an editable selection.

25. (Previously Presented) The computer-readable medium as recited in claim 23, wherein overriding the selection function includes enabling the extensible electronic document editor to interact with the customized selection model without exposing details of the designer.

26. (Original) The computer-readable medium as recited in claim 23, wherein the utilizing the parameters further comprises utilizing the parameters to add a segment to an editable selection.

27. (Original) The computer-readable medium as recited in claim 23, wherein the utilizing the parameters further comprises utilizing the parameters to remove a segment from an editable selection.

28. (Original) The computer-readable medium as recited in claim 23, wherein the utilizing the parameters further comprises utilizing the parameters to set a selection type.

29.-35. (Canceled)

36. (Previously Presented) A method for providing selection services to one or more extensions in an extensible editor, the method comprising:

receiving a request from an extension to utilize a selection services component;

facilitating the request by presenting a selection services interface that is accessible by the extension and that overrides a selection function provided by the extensible editor to provide a customized model for the selection function; and

communicating with the extension through the selection services interface to enable the extension to clear, add, or remove a selected segment from a selection object of an electronic document using the customized model for the selection function.

37. (Original) The method as recited in claim 36, wherein the selection services interface further comprises a method to add an element segment to an editable selection, and wherein the communicating further comprises communicating with the extension through the selection services interface.

38. (Previously Presented) The method as recited in claim 36, wherein the selection services interface is adapted to enable the extensible electronic document editor to interact with the customized selection model without exposing details of the extension.

39. (Original) The method as recited in claim 36, wherein the selection services interface further comprises a method to add a segment to an editable selection, and wherein the communicating further comprises communicating with the extension through the selection services interface.

40. (Original) The method as recited in claim 36, wherein the selection services interface further comprises a method to remove a segment from an editable selection, and wherein the communicating further comprises communicating with the extension through the selection services interface.

41. (Original) The method as recited in claim 36, wherein the selection services interface further comprises a method to set a selection type, and wherein the communicating further comprises communicating with the extension through the selection services interface.

42.-60. (Canceled)

(9) Evidence appendix. None

(10) Related proceedings appendix. None